American Journal of Engineering Research (AJER)
e-ISSN : 2320-0847 p-ISSN : 2320-0936
Volume-03, Issue-03, pp-56-64
www.ajer.org

Research Paper Open Access

An Approach To Design A Controlled Multi-logic Function Generator By Using COG Reversible Logic Gates

Shefali Mamataj[1], Biswajit Das[2]
[1](Department of ECE, Murshidabad College of Engineering and Technology, India)
[2](Department of CSE , Murshidabad College of Engineering and Technology ,India)

***Abstract*: -** In today's world everyday a new technology which is faster, smaller and more complex than its predecessor is being developed. Reversible computation is a research area characterized by having only computational models that is both forward and backward deterministic. Reversible Logic is gaining significant consideration as the potential logic design style for implementation in modern nanotechnology and quantum computing with minimal impact on physical entropy. It has become very popular over the last few years since reversible logic circuits dramatically reduce energy loss. It consumes less power by recovering bit loss from its unique input-output mapping. This paper represents the implementation of conventional Boolean functions for basic digital gate by using COG reversible gate. This paper also represents a multi logic function generator circuit for generating multiple logical function simultaneously using COG gates. And also represents a controlled multi logic function generator circuit for generating any specified output in a controlled way.

Keywords: - *Reversible logic, Basic Reversible Gates, Boolean Function, Logical Operation, Garbage, Constant input , Quantum cost .*

I. INTRODUCTION

Modern digital circuits offer a great deal of computation. As technology evolves and many more transistors can fit in a given area, the concern for power dissipation as heat arises. Reversible logic was first related to energy when gates in 1973.It was Landauer states that information loss due to function irreversibility leads to energy dissipation in 1961who stated that there is small amount of heat dissipation the circuit due to loss of one bit of information and it would be equal to kTln2 where 'k' is Boltzman constant and T is the temperature [1]. This principle is further supported by Bennett that zero energy dissipation can be achieved only when the circuit contains reversible proved by Bennett that the energy kTln2 would not be dissipate from the circuit if input can be extracted from output and it would be possible if and only if reversible gates are used[2]. According to Moore's law the numbers of transistors will double every 18 months. Thus energy conservative devices are the need of the day. The amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. Reversible circuits are those circuits that do not lose information A circuit will be reversible if input vector can be specifically retrieved from output vectors and here is one to one correspondence between input and output [3]. Younis and Knight [4] showed that some reversible circuits can be made asymptotically energy-lossless if their delay is allowed to be arbitrarily large. A reversible logic circuit should have the following features [5]: Use minimum number of reversible gates, Use minimum number of garbage outputs, Use minimum constant inputs.

II. REVERSIBLE LOGIC

A. Definition

Some of the basic Definitions [6] Pertaining to Reversible Logic are
Definition 1: Reversible Logic Function

A reversible logic function is a function which maps each input vector to a unique output vector. A function is said reversible if, given its output, it is always possible to determine back its input, which is the case when there is a one-to-one relationship between input and output states.

Definition 2: Reversible Logic Gate
A reversible logic gate is a device which performs such a one to one mapping. If a reversible logic gate has N inputs, then to perform one to one mapping, the number of outputs should also be N. Then this device is said to be an NxN reversible logic gate. The inputs are denoted by I_1 I_2 I_3 ... I_N and the outputs are denoted by O_1O_2 $O_3...O_N$.
Definition 3: Garbage
These are the outputs that are not used in the synthesis of a function. These may appear to be redundant but are very essential for preserving the reversibility of a gate. It is denoted by GO.
Definition 4: Constant Inputs
These are the inputs that have to be maintained at either a constant 0 or at constant 1 in order to generate a given logical expression using the reversible logic gates. It is abbreviated as CI.
Definition 5: Quantum Cost
This refers to the cost of the circuit in terms of the cost of a primitive gate. It is computed knowing the number of primitive reversible logic gates (1x1 or 2x2) required to realize the circuit. It is denoted as QC.

Definition 6: Gate Count
This refers to the number of gates that are present in the given reversible logic circuit. It is denoted by GC. Another parameter that can be defined in relation to the gate count is the flexibility, which can be defined as the ability of a reversible logic gate in realizing more functions. Higher the flexibility of a gate, lesser is the number of gates that are needed to implement a given function, lesser is the gate count.
Definition 7: Hardware Complexity
The hardware complexity [7] is measured by counting the number of AND operations, number of EX-OR operations and number of OR operations.
Let α = No. of EX-OR operation, β = No. of AND operations , δ = No. of NOT operations
Then the total hardware complexity T is given as sum of EX-OR, AND and NOT operations..

B. Reversible Logic Gate

The important basic reversible logic gates are, Feynman gate [8] which is the only 2x2 reversible gate which is as shown in the figure.1a and it is used most popularly by the designers for fan-out purposes. There is also a double Feynman gate [9],Fredkin gate [10] and Toffoli gate [11],New Gate[12] , Peres gate[13] , all of which can be used to realize important combinational functions and all are 3x3 reversible gates and are as shown in the figure.1b to figure.1e .

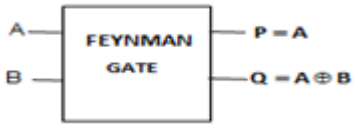

Fig 1a.Feyman gate-

Fig 1b.Fredkin gate-

Fig 1c. Toffoli gate-

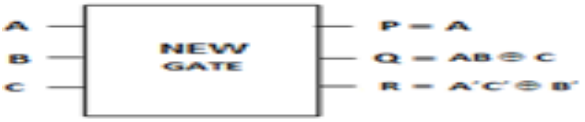

Fig 1d. Newi gate-

Fig 1e. Peres gate-

C. COG reversible gate

A 3X3 reversible gate COG (Controlled Operation Gate) logic already had been proposed [14] shown in Figure 3. The Truth table for the corresponding gate is shown in Table I also .The closer look at the Truth Table reveals that the input pattern corresponding to a specific output pattern can be uniquely determined and thereby maintaining that there is a one-to-one correspondence between the input vector and the output vector. In this gate the input vector is given by IV= (A, B, C) and the corresponding output vector is OV= (P, Q, R)

.

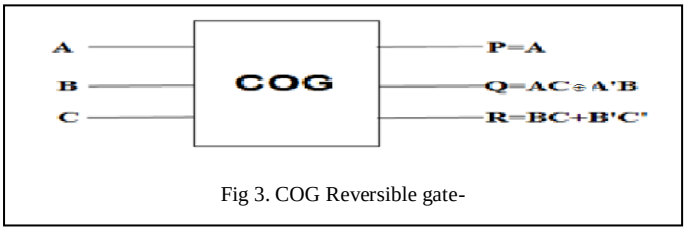

Fig 3. COG Reversible gate-

TableI. Truth table of COG Reversible gate

INPUTS			OUTPUTS		
A	**B**	**C**	**P**	**Q**	**R**
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	1	1

III. IMPLEMENTATION OF CONVENTIONAL BOOLEAN FUNCTION FOR BASIC DIGITAL GATES

We can implement the conventional digital gates by using the COG reversible gate. We can implement the AND, NOT,NAND, NOR, EXOR, EXNOR, OR and COPYING operation which are shown in fig 4a to 4f.In figure 4a We can see that by making the inputs A=A,B=0 and C=B of COG gate we will get AND, NOT & COPYING operation from the output lines. In this way we can get all the operation by setting the input values as per the requirement.

Fig 4a.Implementation of AND,NOT & COPYING gate by COG Reversible gates

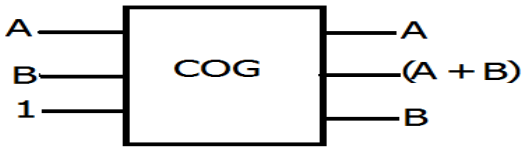

Fig 4b.Implementation of OR & COPYING gate by COG Reversible gates

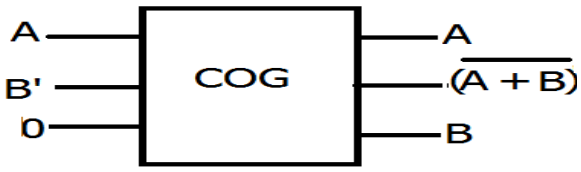

Fig 4c.Implementation of NOR & COPYING gate by COG Reversible gates

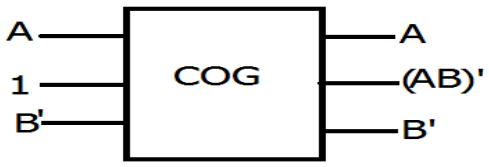

Fig 4d.Implementation of NAND,NOT & COPYING gate by COG Reversible gates

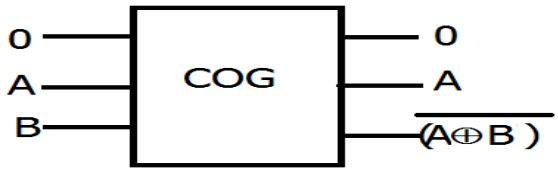

Fig 4e.Implementation of EXNOR & COPYING gate by COG Reversible gates

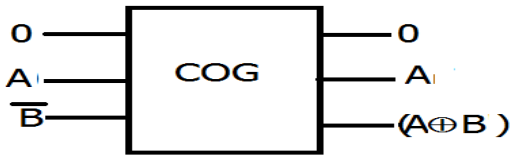

Fig 4f.Implementation of EXOR & COPYING gate by COG Reversible gates

IV. IMPLEMENTATION OF MULTI-LOGIC FUNCTION GENERATOR

A. Multi logic function generator for generating 8 functions

We have implemented a Multi-Logic Function Generator by using COG gates shown in figure 5.For this we have used seven COG reversible gates and one FEYNMAN gate. There are total eight operations at the output .In this way we can generate multiple logical function from this function generator. If two variables A and B are taken. Then the functions which are available AND operation(AB),NOR operation(A+B)',OR operation(A+B), NAND operation(AB)',EX NOR operation(A⊕B)',EXOR operation(A ⊕B),Copying operation(A),NOT operation operation(B')

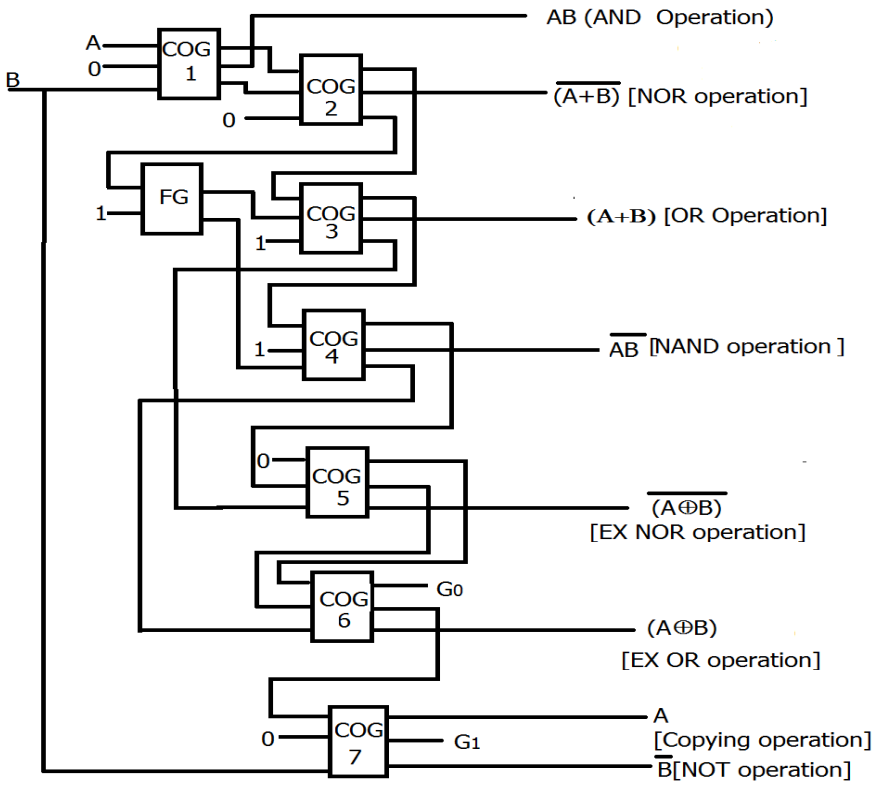

Fig 5.Implementation of MULTI –LOGIC FUNCTION GENERATOR by using COG Reversible gate for generating 8 functions

B. Multi logic function generator for generating 16 functions

We have implemented a Multi-Logic Function Generator by using COG gates shown in figure 6.For this we have used twelve COG reversible gates .There are total sixteen operations at the output .In this way we can generate multiple logical functions from this function generator. If two variables A and B are taken. Then the functions which are available at the output are AND operation(AB),NOR operation(A+B)',OR operation (A+B), NAND operation (AB)',EX-NOR operation (A⊕B)',EXOR operation(A⊕B), Transfer operation (A),

Transfer operation(B), Complement of B operation (B'),Complement of A operation(A'),A inhibits B(AB'), B inhibits A(A'B),B implies A(A+B'),A implies B(A'+B),Identity operation(1),Null operation(0).

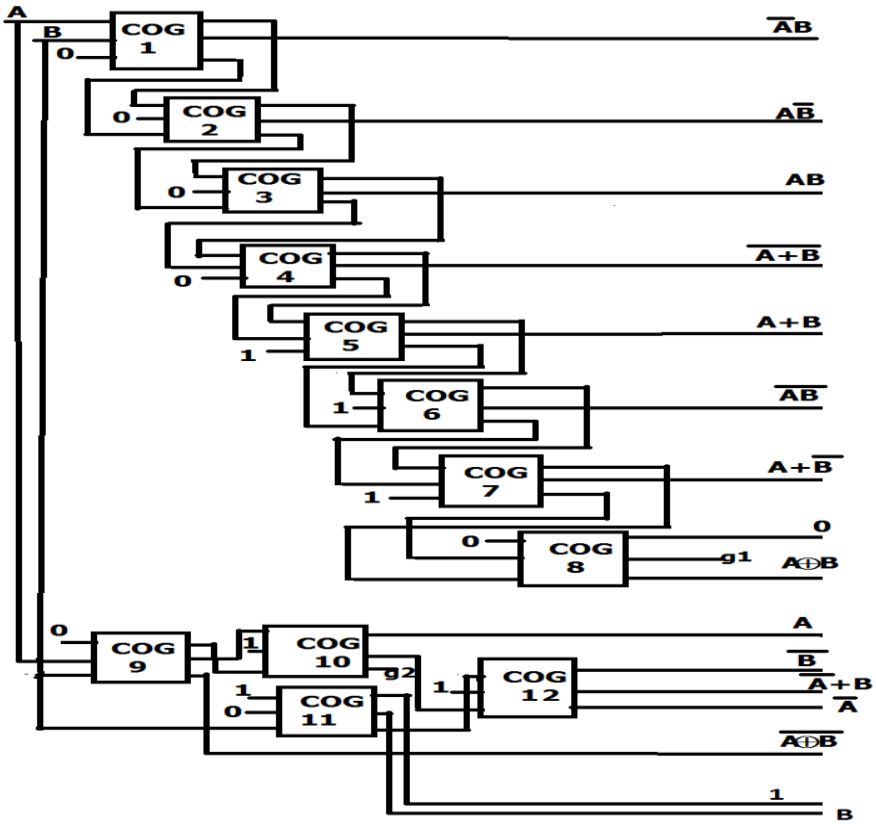

Fig 6.Implementation of MULTI –LOGIC FUNCTION GENERATOR by using COG Reversible gate for generating 16 functions

V. DESIGN OF COMBINATIONAL CIRCUIT

we can design many combinational circuit by using the reversible gates COG, one of these circuit is multiplexer. We can implement 8x1 MUX which is shown in fig7.Generally we use multiplexer for multiplexing one channel out of many channels. For this we have used seven COG reversible gates. There are three selection linesS_3,S_2,S_1 which can be used for channelizing the required input lines. There are total eight input lines I_0,I_1,I_2,I_3,I_4,I_5,I_6,I_7.Out of which we will get any one at the output depending on the selection lines value. There are 10 garbage output.

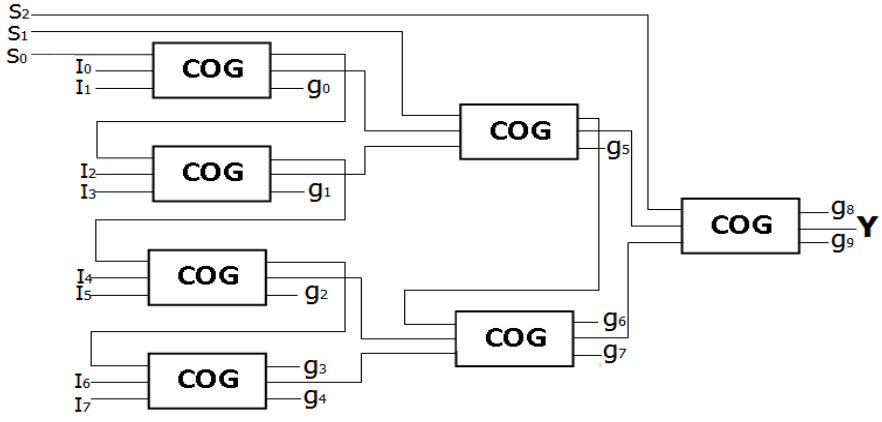

Fig 7.Implementation of 8x1 Multiplexer by COG Reversible gates

We can also implement 16x1 MUX shown in figure 8. For this we have used 15 COG reversible gates. There are four selection lines S_4,S_3,S_2,S_1 which can be used for channelizing the required input lines. There are total sixteen input lines $I_0,I_1,I_2,I_3,I_4,I_5,I_6,I_7I_8,I_9,I_{10},I_{11},I_{12},I_{13},I_{14}I_{15}$.Out of which we will get any one at the output depending on the selection lines value. There are 19 garbage outputs .

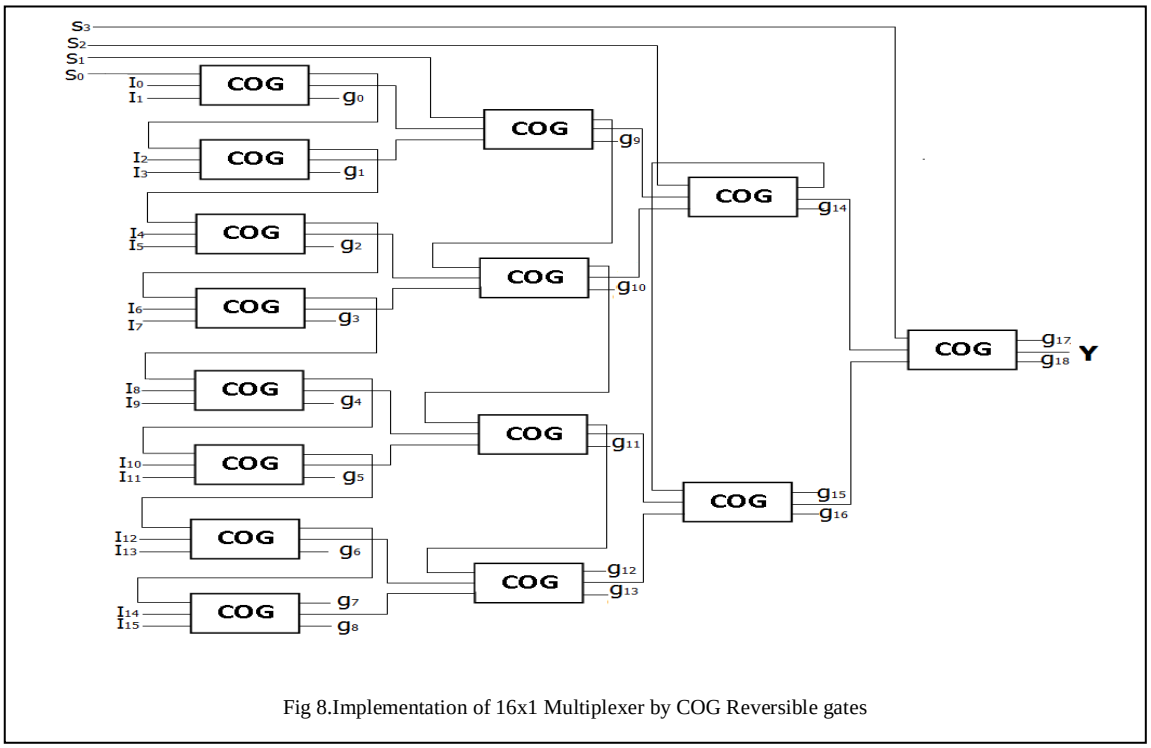

Fig 8.Implementation of 16x1 Multiplexer by COG Reversible gates

VI. IMPLEMENTATION OF PROPOSED CONTROLLED MULTI-LOGIC FUNCTION GENERATOR

A. Controlled Multi logic function generator for generating 8 functions

We have cascaded the function generator with the 8x1 multiplexer to get the controlled output shown in figure 9.Depending upon the value of the selection lines we will get the logical function at the output of the multiplexer which has already been specified.

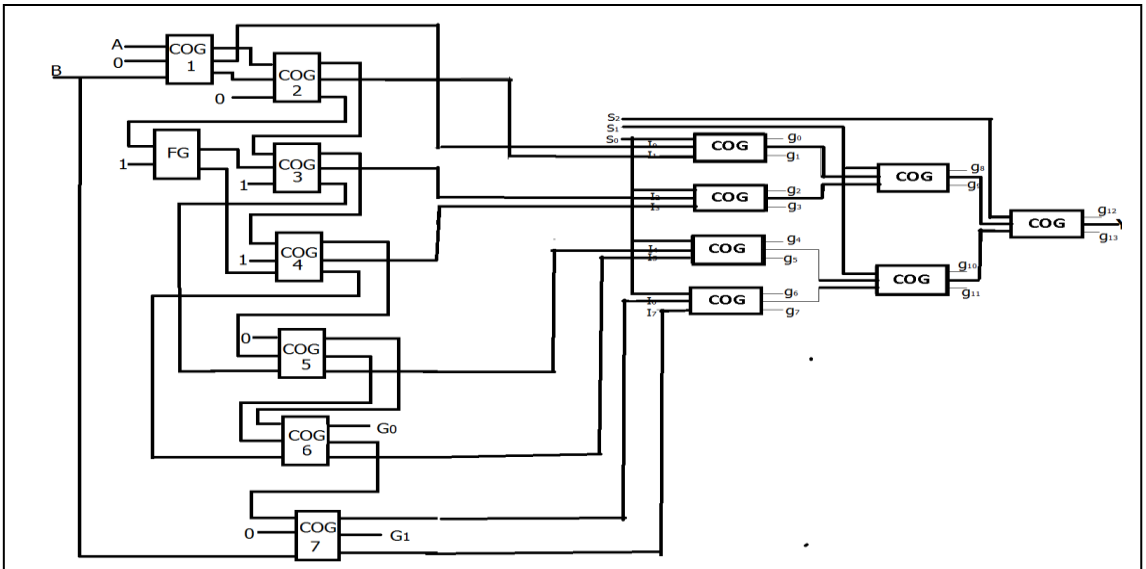

Fig 9.Implementation of controlled multi logic function generator using 8x1 Multiplexer by COG Reversible gates

So we can get the controlled behavior at the output of the circuit depending upon the values of S3, S2, and S1shown in TableII.
Table II.Variation at Output of 8x1 MUX

S3 S2 S1	Boolean function	Name of the function	Output Y of 8x1 MUX
0 0 0	(AB)	AND operation	I_0
0 0 1	(A+B)'	NOR operation	I_1
0 1 0	(A+B)	OR operation	I_2
0 1 1	(AB)'	NAND operation	I_3
1 0 0	(A⊕B)'	EX NOR operation	I_4
1 0 1	(A ⊕B)	EXOR operation	I_5
1 1 0	(A)	Copying operation	I_6
1 1 1	(B')	NOT operation	I_7

B. Controlled Multi logic function generator for generating 16 functions

We have also cascaded the function generator of 16 functions with the 16x1 multiplexer to get the controlled output shown in figure 10.Depending upon the value of the selection lines we will get the logical function at the output of the multiplexer which has already been specified by the selection values.

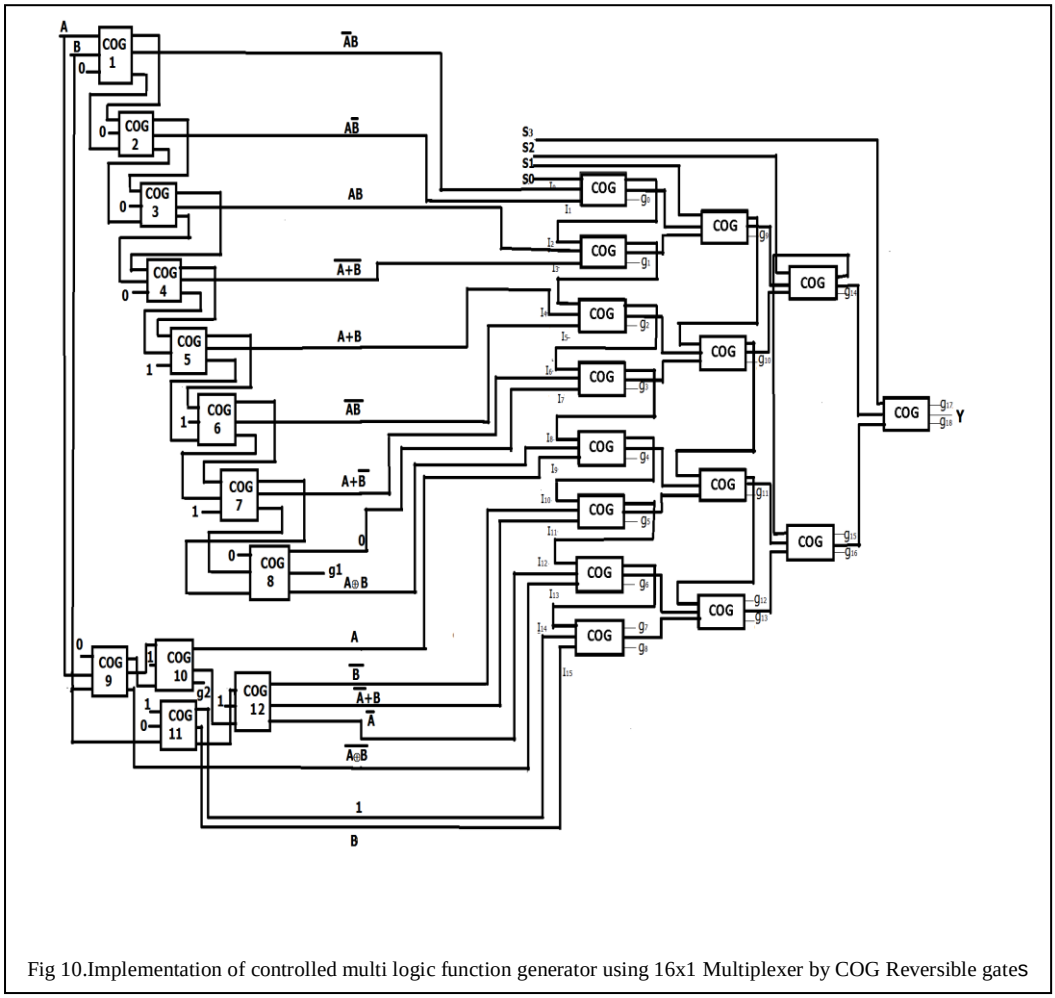

Fig 10.Implementation of controlled multi logic function generator using 16x1 Multiplexer by COG Reversible gateS

So we can get the controlled behavior at the output of the circuit depending upon the values ofS4, S3, S2, and S1shown in Table III.

Table III. Variation at Output of 16x1 MUX

S4 S3 S2 S1	Boolean function	Name of the function	Output Y of 16x1 MUX
0 0 0 0	A'B	B inhibits A	I_0
0 0 0 1	AB'	A inhibits B	I_1
0 0 1 0	AB	AND	I_2
0 0 1 1	(A+B)'	NOR	I_3
0 1 0 0	(A+B)	OR	I_4
0 1 0 1	(AB)'	NAND	I_5
0 1 1 0	A+B'	B implies A	I_6
0 1 1 1	0	Null	I_7
1 0 0 0	(A⊕ B)	EX OR	I_8
1 0 0 1	A	Transfer A	I_9
1 0 1 0	B'	Complement of B	I_{10}
1 0 1 1	A'+B	A implies B	I_{11}
1 1 0 0	A'	Complement of A	I_{12}
1 1 0 1	(A⊕ B)'	EX NOR	I_{13}
1 1 1 0	1	Identity	I_{14}
1 1 1 1	B	Transfer B	I_{15}

VII. COMPARISONS RESULTS

We have compared the proposed function generators in terms of number of gates,constant input ,garbage output and and hardware complexity shown in Table IV .

Parameters to be campared	Function Generator of 8 Functions		Function Generator of 16 Functions	
	Without Controlled Circuit	With Controlled Circuit	Without Controlled Circuit	With Controlled Circuit
Number of gates	8	15	12	27
Constant input	7	7	13	13
Garbage output	2	12	2	21
Hardware complexity	15α+14ß+14 δ	29α+28ß+28 δ	24 α+24ß+24 δ	54α+54ß+54 δ

VIII. CONCLUSIONS

We have compared the function generators in terms of number of gates, constant input, garbage output and hardware complexity shown in table IV. These proposed function generators can produce different logical functions. The 8 multi logic functions generator generates 8 logical functions whereas the 16multi logic function generator generates 16 functions. Most of the other function generator in the literature aim at producing only the main functions such as AND,OR,EX-OR and EX-NOR and use these functions to produce the remaining functions The distinguishing feature of these designs are that all the functions are produced within a single unit and the user can choose any one of the particular functions by the help of control unit .These Function generators can be efficiently used in the designing of ALU.As a future work there is vast application of these proposed designs and we would like to implement these design using quantum dot cellular automata.

IX. ACKNOWLEDGEMENTS

The authors wish to thank ECE Department and CSE Department of Murshidabad College of Engineering and Technology, Berhampore for supporting this work.

REFERENCES

[1] R.Landauer,-Irreversibility and Heat Generation in the Computational Process, IBM Journal of Research and Development , 5, pp. 183-191,1961 .

[2] C.H.Bennett, Logical Reversibility of Computation‖, IBM J.Research and Development, pp. 525-532, November 1973 .

[3] Pradeep singla and Naveen kr. Malik " A Cost - Effective Design of Reversible programmable logic array "International Journal Of Computer Application , volume 41 – no. 15, march- 2012.

[4] S.Younis and T. Knight, "Asymptotically Zero Energy Split Level Charge Recovery Logic, "Workshop on Low Power Design, June1994.
[5] Perkowski, M. and P. Kerntopf, "Reversible Logic. Invited tutorial" Proc. EURO-MICRO,Warsaw, Poland ,Sept 2001.
[6] Rakshith Saligram and Rakshith T R "Novel Code Converter employing Reversible Logic" Intl. Journal of Computer Applications, Vol 52, No. 18, Aug 2012.
[7] Md. Saiful Islam et.al" Synthesis of fault tolerant Reversible logic" IEEE 2009.
[8] R.Feynman, "Quantum Mechanical Computers", Optical News, pp. 11-20, 1985
[9] B.Parhami; "Fault Tolerant Reversible Circuits" Proc. 40th Asilomar Conf. Signals, Systems, and Computers, Pacific Grove,CA, Oct.2006.
[10] Fredkin, T. Toffoli, "Conservative Logic", International Journal of Theory. Physics, 21,pp.219-253, 1982.
[11] T.Toffoli.,"Reversible Computing", Tech memo MIT/LCS/TM - 151,MIT Lab for Computer Science (1980).
[12] Md. M. H Azad Khan, "Design of Full-adder With Reversible Gates", International Conference on Computer and Informa tion Technology, Dhaka,Bangladesh,pp.515-519,2002.
[13] Peres, A.. Reversible logic and quantum computers, Physical Review: A, 32 (6): 3266- 3276, 1985
[14] Shefali Mamataj, Biswajit Das, Anurima Rahaman "An Ease implementation of 4-bit Arithmetic Circuit for 8 Operation by using a new reversible COG gate"International Journal of Advanced Research in Electrical, Electronics and Instrumen tation EngineeringVol. 3, Issue 1, January 2014.